

# 华东师范大学计算机科学技术系上机实践报告

课程名称：计算机网络

年级：2022

上机实践成绩：

指导教师：洪道诚

姓名：朱宇笑

创新实践成绩：

实验名称：地址解析协议ARP

学号：10225001410

上机实践日期：2023/12/2

座位编号：F

组号：6

上机实践时间：2 学时

## 一、 实验目的

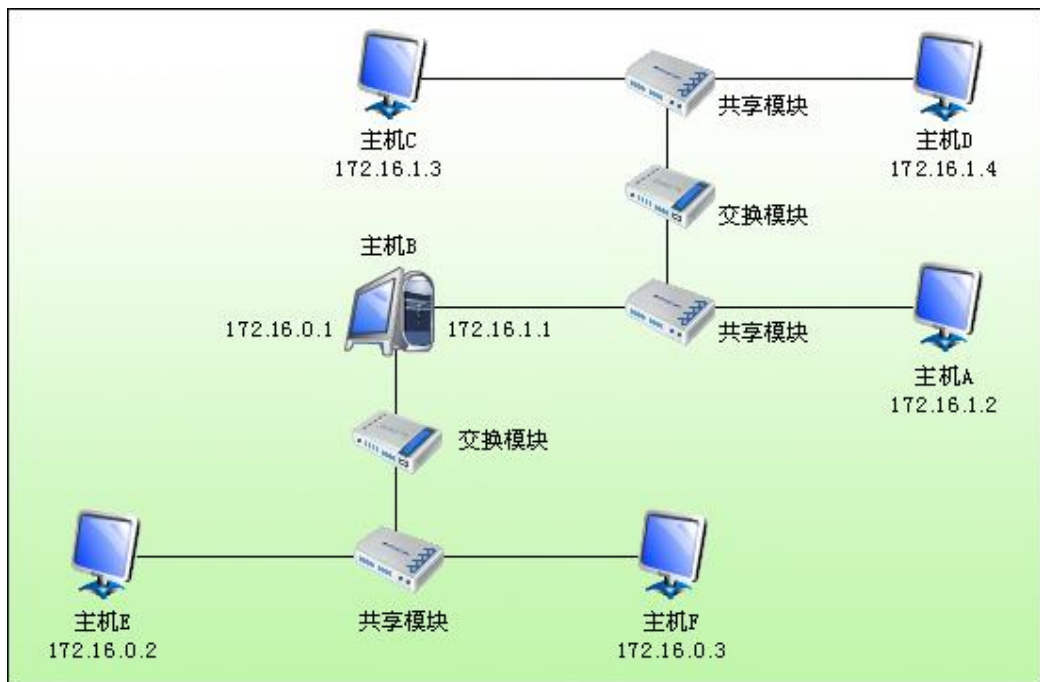
1. 掌握ARP协议的报文格式
2. 掌握ARP协议的工作原理
3. 理解ARP高速缓存的作用
4. 掌握ARP请求和应答的实现方法
5. 掌握ARP缓存表的维护过程

## 二、 实验设备

1. PC机
2. 仿真编辑器
3. 协议分析器

## 三、 实验原理

本实验采用网络结构二



## 一. 物理地址与逻辑地址

### 1. 物理地址

物理地址是节点的地址，由它所在的局域网或广域网定义。物理地址包含在数据链路层的帧中。物理地址是最低一级的地址。

物理地址的长度和格式是可变的，取决于具体的网络。以太网使用写在网络接口卡（NIC）上的 6 字节的标识作为物理地址。

物理地址可以是单播地址（一个接收者）、多播地址（一组接收者）或广播地址（由网络中的所有主机接收）。有些网络不支持多播或广播地址，当需要把帧发送给一组主机或所有主机时，多播地址或广播地址就需要用单播地址来模拟。

### 2. 逻辑地址

在互联网的环境中仅使用物理地址是不合适的，因为不同网络可以使用不同的地址格式。因此，需要一种通用的编址系统，用来惟一地标识每一台主机，而不管底层使用什么样的物理网络。

逻辑地址就是为此目的而设计的。目前 Internet 上的逻辑地址是 32 位地址，通常称为 IP 地址，可以用来标识连接在 Internet 上的每一台主机。在 Internet 上没有两个主机具有同样的 IP 地址。

逻辑地址可以是单播地址、多播地址和广播地址。其中广播地址有一些局限性。在实验三中将详细介绍这三种类型的地址。

## 二. ARP 协议简介

Internet 是由各种各样的物理网络通过使用诸如路由器之类的设备连接在一起组成的。主机发送一个数据包到另一台主机时可能要经过多种不同的物理网络。主机和路由器都是在网络层通过逻辑地址来识别的，这个地址是在全世界范围内是惟一的。然而，数据包是通过物理网络传递的。在物理网络中，主机和路由器通过其物理地址来识别的，其范围限于本地网络中。物理地址和逻辑地址是两种不同的标识符。这就意味着将一个数据包传递到一个主机或路由器需要进行两级寻址：逻辑地址和物理地址。需要能将一个逻辑地址映射到相应的物理地址。

ARP 协议（地址解析协议）是“AddressResolutionProtocol”的缩写。所谓“地址解析”就是主机在发送帧前将目的逻辑地址转换成目的物理地址的过程。在使用 TCP/IP 协议的以太网中，ARP 协议完成将 IP 地址映射到 MAC 地址的过程。

## 三. ARP 报文格式

下图为 ARP 数据报的报文格式：

硬件类型（16 位）		协议类型（16 位）
硬件地址长度 （8 位）	协议地址长度 （8 位）	操作码（16 位）
发送端硬件地址 （例如，对以太网是 6 字节）		

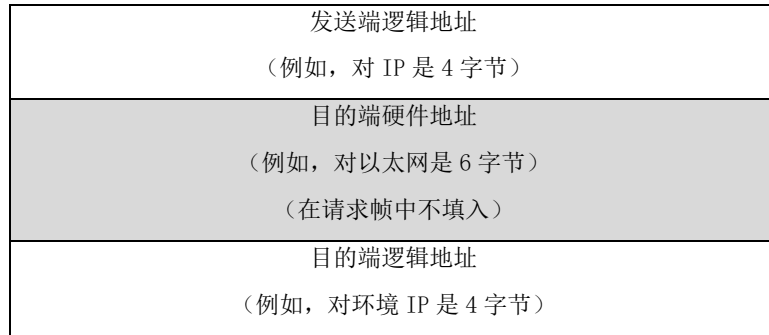


图 2-1 ARP 报文格式

ARP 报文格式具有如下的一些字段:

- **硬件类型:** 这是 16 位字段, 用来定义运行 ARP 的网络的类型。每一个局域网基于其类型被指派给一个整数。例如, 以太网的硬件类型是 1。ARP 可用在任何网络上。
- **协议类型:** 这是 16 位字段。用来定义协议的类型。例如, 对 IPv4 协议, 这个字段的值是 0x0800。ARP 可用于任何高层协议。
- **硬件地址长度:** 这是一个 8 位字段, 用来定义以字节为单位的物理地址长度。例如, 以太网物理地址为 6 字节, 所对应的硬件地址长度值为 6。
- **协议地址长度:** 标识用于该数据包的逻辑地址的长度, 用十进制标识, 单位为一个字节, 例如, IPv4 为 4 个字节, 所对应的协议地址长度值为 4。
- **操作码:** 这是 16 位字段, 用来定义数据包的类型。已定义了两种类型: 为 1 时表示 ARP 请求, 为 2 时表示 ARP 应答。
- **发送端硬件地址:** 这是可变长度字段, 用来定义发送端的物理地址。对于以太网这个字段是 6 字节长。
- **发送端逻辑地址:** 这是可变长度字段, 用来定义发送端的逻辑地址。对于逻辑地址为 IP 地址的网络, 该字段长度为 4 字节。
- **目的端硬件地址:** 这是可变长度字段, 用来定义目标的物理地址。对于 ARP 请求, 字段是全 0, 因为发送端不知道目标的物理地址 (该字段长度为可变, 如以太网硬件地址为 6 个字节)。
- **目的端逻辑地址:** 这是可变长度字段, 用来定义目标的逻辑地址 (该字段长度为可变, 如 IPv4 协议的逻辑地址为 4 个字节)。

#### 四. ARP 封装

ARP 数据报直接封装在数据链路帧中。例如, 在下图中, ARP 数据包封装在以太网的帧中。类型字段值为 0x0806 指出了此帧所携带的数据是 ARP 数据包。

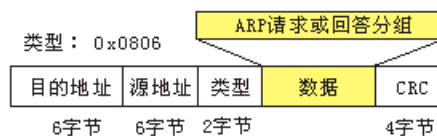


图 2-2 ARP 数据包的封装

## 五. ARP 的运行过程

数据包传输过程可分为如下步骤：

1. 发送端知道目的端的 IP 地址。
2. IP 要求 ARP 创建一个 ARP 请求报文，其中包含了发送方的物理地址、发送方的 IP 地址和目的端的 IP 地址。目的端的物理地址用 0 填充。
3. 将报文传递到数据链路层，并在该层中用发送方的物理地址作为源地址，用物理广播地址作为目的地地址，将其封装在一个帧中。
4. 因为该帧中包含了一个广播目的地地址，所以同一链路中的每个主机或路由器都接收到这个帧。所有接收到该帧的主机都将其传递到 ARP 层进行处理。除了目的端主机以外的所有主机都丢弃该报文。
5. 目的端主机用一个包含其物理地址的 ARP 应答报文做出响应，并对该报文进行单播。
6. 发送方接收到这个应答报文，这样它就知道了目标主机的物理地址。

ARP 地址解析过程如下图所示。

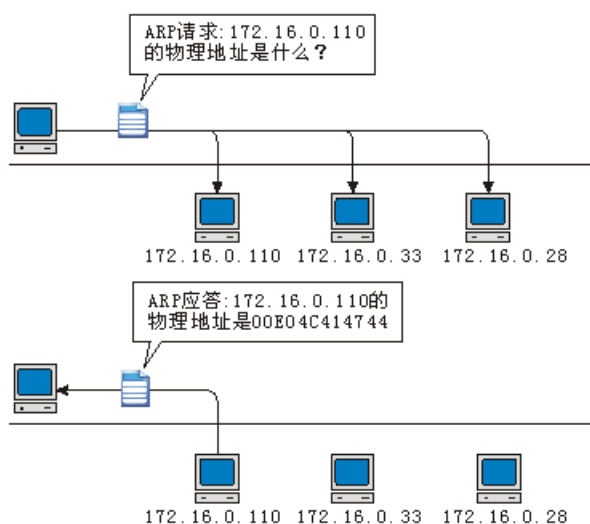


图 2-3ARP 地址解析过程

## 六. ARP 高速缓存

在真正的协议实现中，并不是每次发送 IP 报文前都需要发送 ARP 请求报文来获取目的 MAC 地址。在大多数的系统中都存在着一个 ARP 缓存表。记录着一段时间内曾经获取过的 MAC 地址和 IP 地址的映射关系，如下图所示：

IP 地址	MAC 地址
202.98.13.1	00-E0-4C-3D-89-76
202.98.13.2	00-E0-4C-3D-C5-03
202.98.13.3	00-E0-4C-4D-BA-92
.....	.....

图 2-4ARP 高速缓存

发送 IP 数据报前先对 ARP 缓存表进行查找，查看目的 MAC 地址是否存在于缓存表中，如果存在，则

不需要发送 ARP 请求报文而直接使用此地址进行 IP 数据包的发送。如果不存在，则发送 ARP 请求报文，在收到 ARP 应答报文之后，使用应答报文中的目的 MAC 地址发送 IP 数据包，并将目的 MAC 地址存于 ARP 缓存表中供以后使用。

另外，ARP 缓存表采用老化机制，在一段时间内如果表中的某一项没有使用，就会被删除，这样可以大大减少 ARP 缓存表的长度，加快查询速度。

下图描述了 ARP 高速缓存的使用与更新过程：

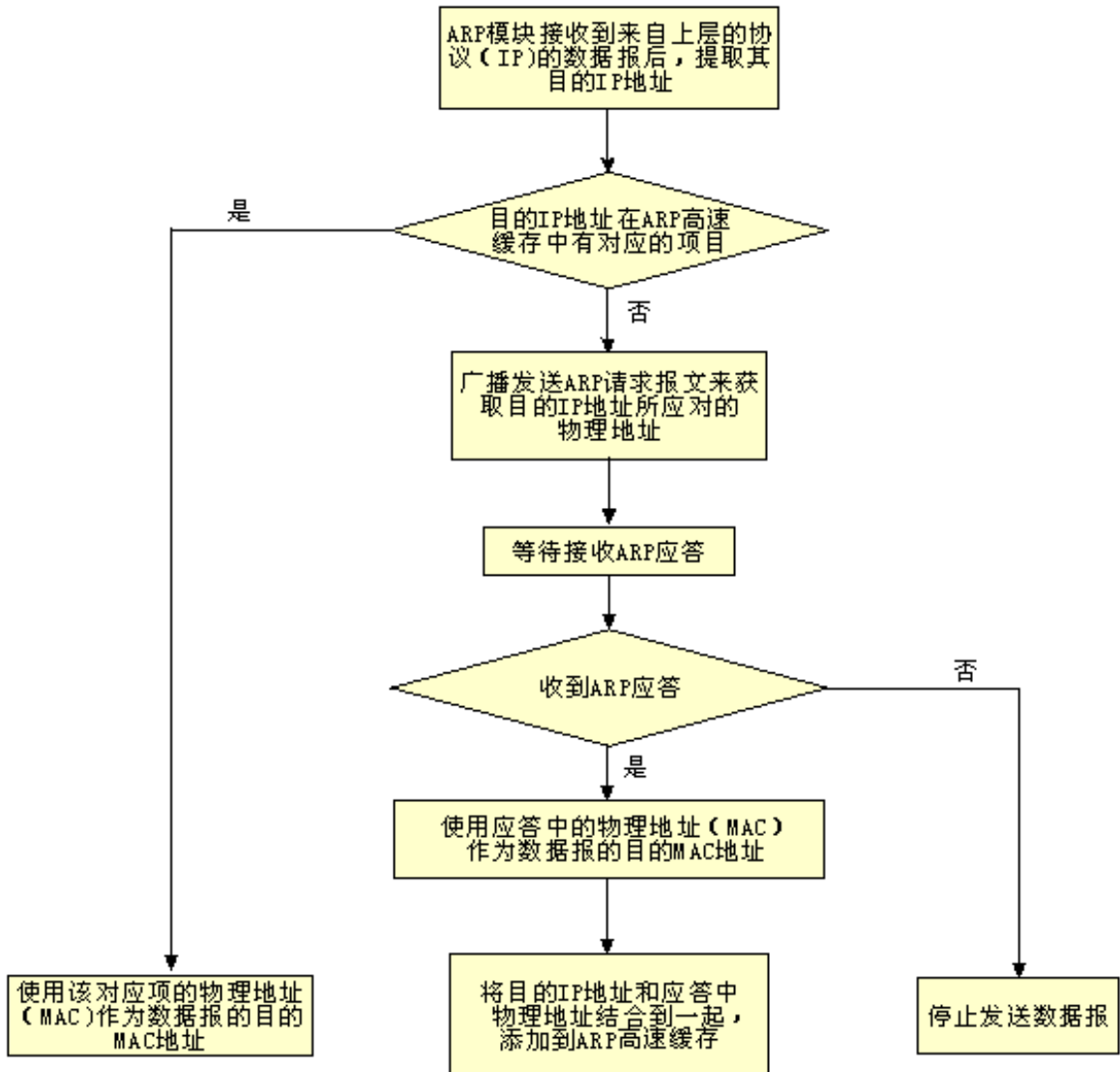


图 2-5 ARP 高速缓存的使用与更新过程

### 七. 代理 ARP

代理 ARP 可用来产生划分子网的效应。如果 ARP 请求是从一个网络中的主机发往另一个网络中的主机，那么连接这两个网络的路由器就可以回答该请求，当这个路由器收到真正的 IP 数据包时，它就把该数据包发送给相应的主机或路由器。

例如，在下图所示的网络中，安装在右边主机上的代理 ARP 应答对目标 IP 地址为 141.23.56.23 的 ARP 请求。

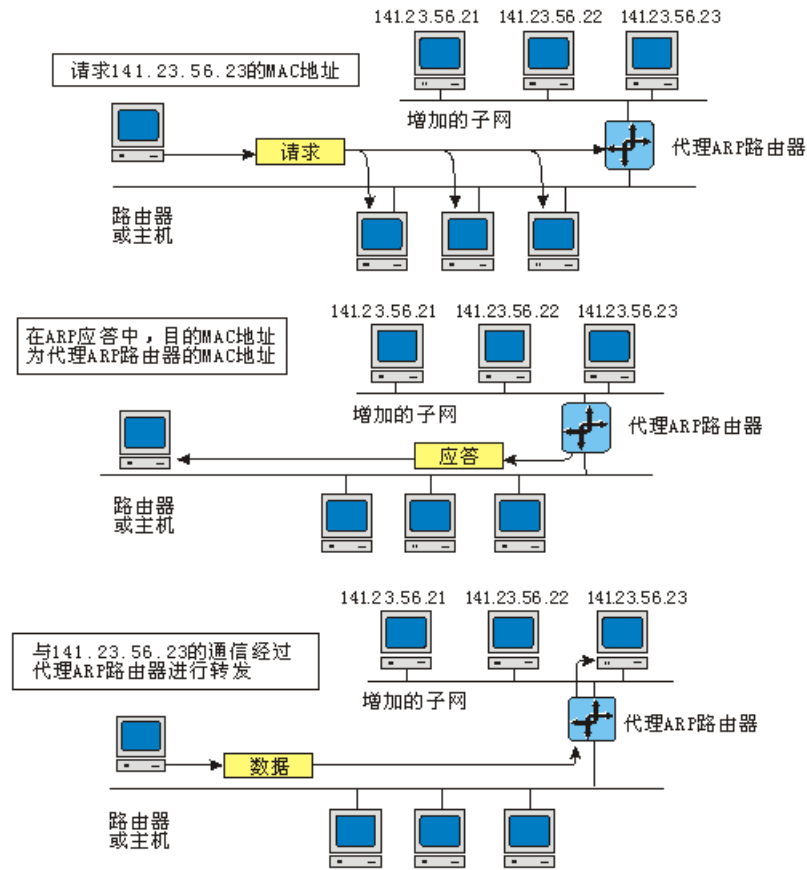


图 2-6 代理 ARP

### 八. 协议栈实现代码解析

本实验将通过安装目录

JLCS\ExpCNC\Work\NPL\ExpNPL\_student\netproto\_arp\_student\netproto\_arp\_student 下的 netproto\_arp\_student.h 和 netproto\_arp\_student.c 两个文件进行编码，完成协议栈中 arp 协议的实现。

netproto\_arp\_student.h 文件中定义了 arp 协议实现相关数值，关键代码如下所示：

```
#define MAC_PROTO_ARP          0x0806
#define ARP_HWTYPE_ETH        0x0001
#define ARP_PROTOTYPE_IP      0x0800
#define ARP_HWADDR_LEN_ETH    0x06
#define ARP_PROTOADDR_LEN_IP  0x04
#define ARP_OPCODE_REQUEST    0x0001
#define ARP_OPCODE_RESPONSE   0x0002

#define ARP_DEST_IP           "0.0.0.0"
```

这段代码定义了 8 个宏，他们代表的含义如下表所示：

表 2-1netproto\_arp\_student.h 中定义的宏

宏	值	描述
MAC_PROTO_ARP	0x0806	定义以太网帧中的“协议类型或数据长度”字段的值
ARP_HWTYPE_ETH	0x0001	定义 arp 包头中“硬件类型”字段值
ARP_PROTOTYPE_IP	0x0800	定义 arp 包头中“协议类型”字段值
ARP_HWADDR_LEN_ETH	0x06	定义 arp 包头中“硬件地址长度”字段值
ARP_PROTOADDR_LEN_IP	0x04	定义 arp 包头中“协议地址长度”字段值
ARP_OPCODE_REQUEST	0x0001	定义 arp 请求数据包头中“操作码”字段值
ARP_OPCODE_RESPONSE	0x0002	定义 arp 应答数据包头中“操作码”字段值
ARP_DEST_IP	"0.0.0.0"	用点分十进制表示的 IP 地址，定义 arp 包头中“目的端逻辑地址”字段值

在实验的编码过程中，应该使用这些宏对相应的变量进行赋值。

netproto\_arp\_student.c 文件是协议栈中 arp 协议的实现部分，其中定义了 1 个全局数组以及 3 个函数。下面分别介绍这些协议栈的实现部分。

全局数组 netp\_arp\_table 是 arp 协议的缓存表，拥有 NETP\_ARP\_TABLE\_SIZE 个 netp\_arp\_table\_item 元素。其中 NETP\_ARP\_TABLE\_SIZE 是 arp 缓存表的条目数，默认值为 10，同学们不需要修改。netp\_arp\_table\_item 是一个结构体，代表了 arp 缓存表中的一个条目，包括一个物理地址 hardware\_addr 和一个 IP 地址 ip\_address，其定义如下：

```
struct netp_arp_table_item
{
    u8_t hardware_addr[ETH_ADDRESS_LEN];
    struct ip_addr ip_address;
};
```

需要根据 arp 协议的实现原理编写代码来维护 arp 缓存表。

函数 display\_arp\_table 的功能是显示 arp 缓存表中的条目，在实验中可以直接调用该函数，便于查看 arp 缓存表中的内容。

函数 netp\_arp\_output\_student 的功能是构造并发送一个 arp 请求数据包。这个函数的编码工作需要由学生完成。

当有数据到达本机网络接口时，函数 netp\_arp\_input\_student 将被调用，并传递给这个函数原始数据。在本实验中该函数需要完成两个功能，一是处理针对本机的 arp 请求数据包，二是处理针对本机的 arp 应答数据包。处理针对本机的 arp 请求数据包时，应该发送相应的 arp 应答数据包。处理针对本机的 arp 应答数据包时，应该更新 arp 缓存表。这个函数的编码工作需要由学生完成。该函数的返回值为 push\_to\_lwip 的枚举类型值，push\_to\_lwip 的定义如下：

```

enumpush_to_lwip{
NETP_PUSH_TO_LWIP//数据处理完成后, 交给 lwip 继续处理
NETP_NO_PUSH_LIWP//数据处理完成后, 不交给 lwip 继续处理
//本层处理完毕以后数据包被丢弃
};

```

返回 `NETP_PUSH_TO_LWIP` 表示这个数据帧应该提交给协议栈上层继续处理, 而返回 `NETP_NO_PUSH_LIWP` 则表示不需要提交给协议栈上层处理, 本层处理完毕后, 这个数据帧将被丢弃。需要根据正确的逻辑关系返回适当的值, 使协议栈正常工作。

在编码过程中可能会遇到一些结构体、宏和函数, 下表对他们进行介绍:

表 2-2 实验涉及的结构体和函数

结构体/函数	声明或定义	描述
<code>structnetp_arp_table_item</code>	<pre> structnetp_arp_table_item{ u8_thardware_addr[ETH_ADDRESS_LEN]; structip_addrrip_address;}; </pre>	ARP 缓存表中一个条目的结构
<code>structin_addr</code>	<pre> structin_addr{ u32_ts_addr;}; </pre>	32 位地址
<code>structnetp_eth_header</code>	<pre> structnetp_eth_header{ u8_tdest_address[ETH_ADDRESS_LEN]; u8_tsour_address[ETH_ADDRESS_LEN]; u16_ttype;}; </pre>	以太网帧头结构
<code>structnetp_arp_header</code>	<pre> structnetp_arp_header{ u16_thardware_type; u16_tproto_type; u8_thw_addr_len; u8_tproto_addr_len; u16_topcode; u8_tsrc_hw_addr[ETH_ADDRESS_LEN]; structip_addrsrc_ip_addr; u8_tdest_hw_addr[ETH_ADDRESS_LEN]; structip_addrdest_ip_addr; }; </pre>	arp 协议包头结构
<code>NETP_ARP_TABLE_SIZE</code>	<code>#defineNETP_ARP_TABLE_SIZE10</code>	ARP 缓存表中条目的个数
<code>ETH_ADDRESS_LEN</code>	<code>#defineETH_ADDRESS_LEN6</code>	以太网帧地址长度
<code>ETH_HEADER_LEN</code>	<code>#defineETH_HEADER_LEN14</code>	以太网帧头长度
<code>ARP_HEADER_LEN</code>	<code>#defineARP_HEADER_LEN28</code>	ARP 数据包头长度
<code>display_arp_table</code>	<code>voiddisplay_arp_table();</code>	将 ARP 缓存表的内容显示到标准输出



结构体/函数	声明或定义	描述
netp_current_hw_addr	<code>intnetp_current_hw_addr( u8_t*hardware_address );</code>	获取正在使用的网络适配器的物理地址
netp_current_ip_addr	<code>u32_tnetp_current_ip_addr();</code>	获取当前正在使用的网络适配器的 IP 地址
htons	<code>u16_thtons(u16_tn);</code>	将 16 位数值由主机字节序转换为网络字节序
inet_addr	<code>u32_tinet_addr(constchar*cp);</code>	将 ASCII 编码的 Internet 地址转换成为网络字节序地址
netp_packet_send	<code>intnetp_packet_send( void*buffer, intlen );</code>	使用当前正在使用的网络接口发送一个数据帧

## 九. 各模块推荐流程

### 1. arp 请求发送流程

编码实现 arp 请求数据包发送推荐使用如下流程：

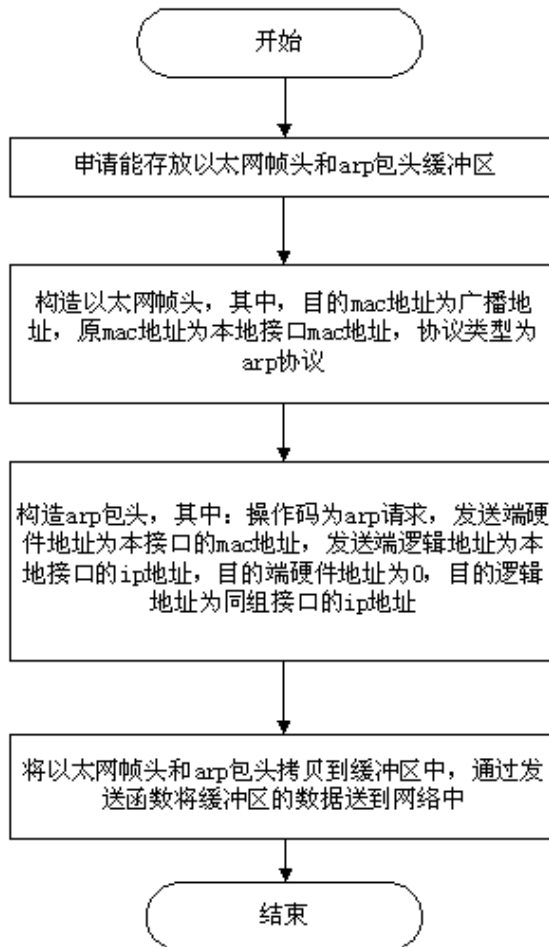


图 2-7arp 请求数据包发送推荐流程

## 2. 输入 arp 数据包处理流程

编码实现处理 arp 输入数据包推荐使用如下流程：

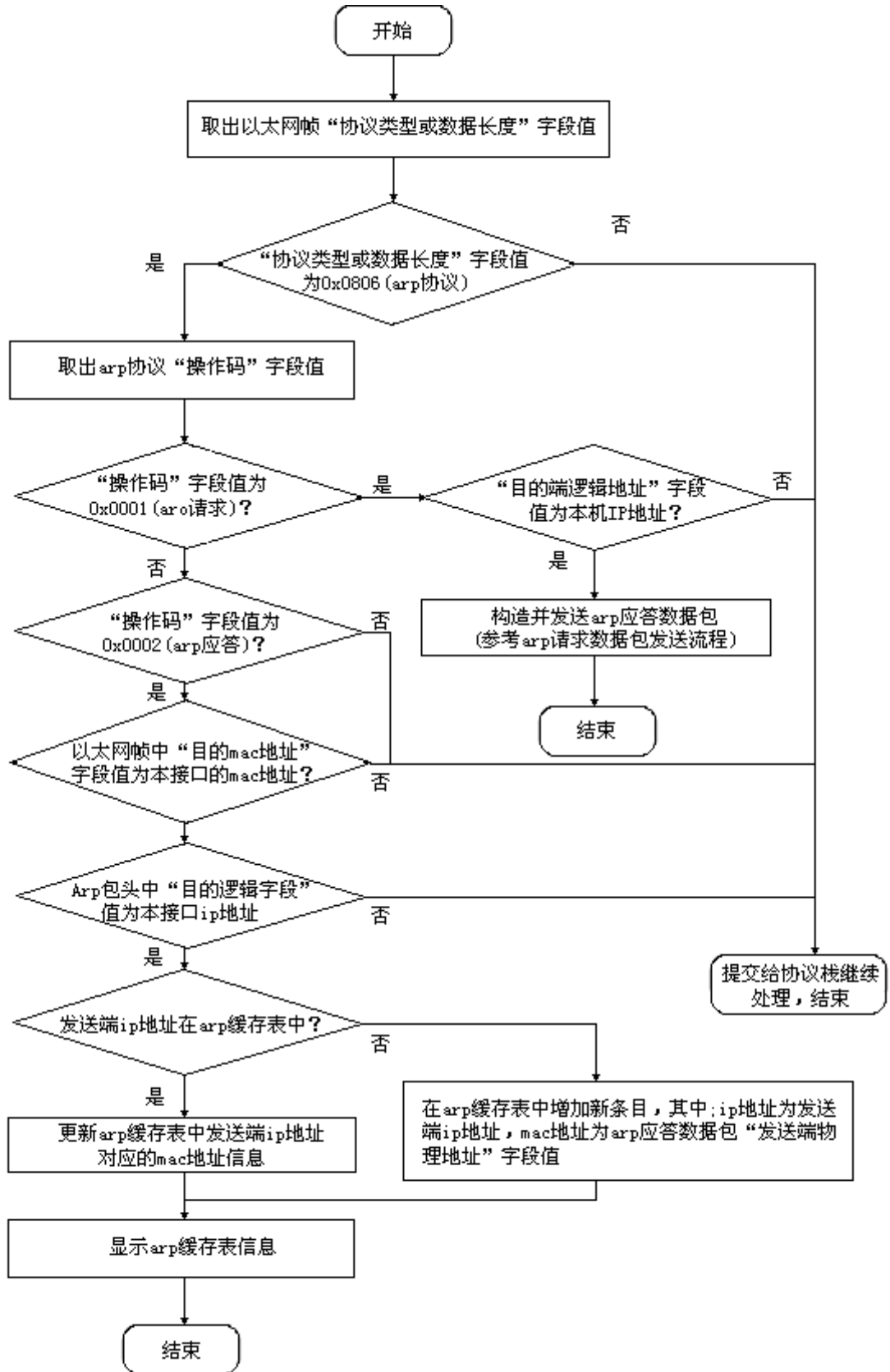


图 2-8 处理 arp 输入数据包推荐流程

## 四、 实验步骤

### 练习 1 领略真实的 ARP（同一子网）

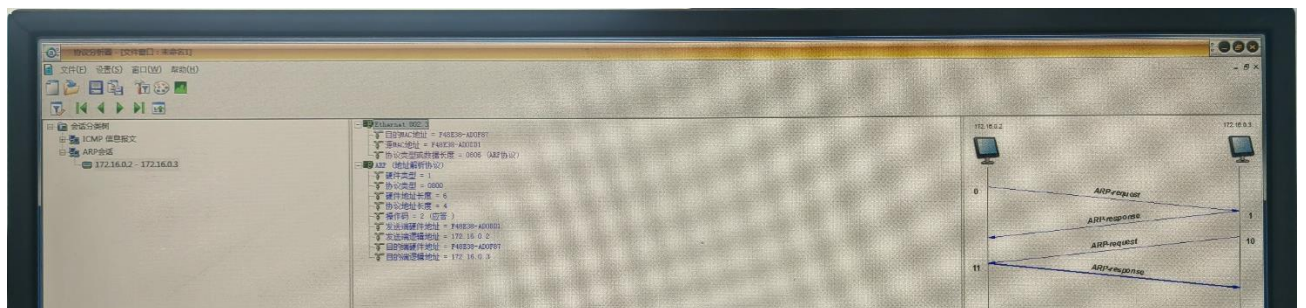
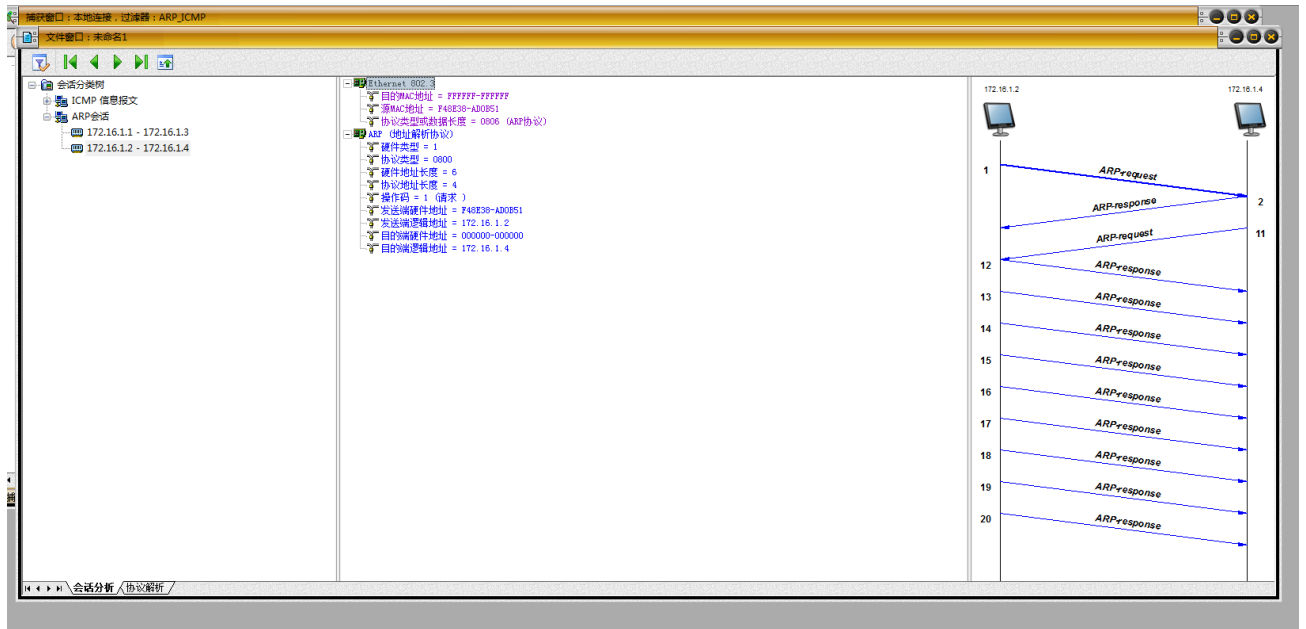
各主机打开工具区的“拓扑验证工具”，选择相应的网络结构，配置网卡后，进行拓扑验证，如果通过拓扑验证，关闭工具继续进行实验，如果没有通过，请检查网络连接。

本练习将主机 A、B、C、D、E、F 作为一组进行实验。

1. 主机 A、B、C、D、E、F 启动协议分析器，打开捕获窗口进行数据捕获并设置过滤条件（提取 ARP、ICMP）。
2. 主机 A、B、C、D、E、F 在命令行下运行“arp -d”命令，清空 ARP 高速缓存。
3. 主机 A ping 主机 D（172.16.1.4）。

主机 B ping 主机 C（172.16.1.3）。

主机 E ping 主机 F（172.16.0.3）。



```

管理员: C:\Windows\system32\cmd.exe

C:\Users\Administrator>arp -d

C:\Users\Administrator>arp -d

C:\Users\Administrator>ping 172.16.1.4

正在 Ping 172.16.1.4 具有 32 字节的数据:
来自 172.16.1.4 的回复: 字节=32 时间=2ms TTL=128
来自 172.16.1.4 的回复: 字节=32 时间=1ms TTL=128
来自 172.16.1.4 的回复: 字节=32 时间=1ms TTL=128
来自 172.16.1.4 的回复: 字节=32 时间=1ms TTL=128

172.16.1.4 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间<以毫秒为单位>:
    最短 = 1ms, 最长 = 2ms, 平均 = 1ms

C:\Users\Administrator>arp -a

接口: 172.16.1.2 --- 0xb
Internet 地址          物理地址          类型
172.16.1.4             f4-8e-38-ad-0d-c4 动态
    
```

```

C:\Users\Administrator>arp -a

接口: 172.16.1.1 --- 0xb
Internet 地址          物理地址          类型
172.16.1.3             f4-8e-38-ad-0d-c2 动态
172.16.1.255          ff-ff-ff-ff-ff-ff 静态
224.0.0.252           01-00-5e-00-00-fc 静态

接口: 192.168.221.1 --- 0xd
Internet 地址          物理地址          类型
192.168.221.255       ff-ff-ff-ff-ff-ff 静态
224.0.0.252           01-00-5e-00-00-fc 静态

接口: 192.168.91.1 --- 0xf
Internet 地址          物理地址          类型
192.168.91.255        ff-ff-ff-ff-ff-ff 静态
224.0.0.252           01-00-5e-00-00-fc 静态

接口: 172.16.0.1 --- 0x11
Internet 地址          物理地址          类型
172.16.0.255          ff-ff-ff-ff-ff-ff 静态
224.0.0.252           01-00-5e-00-00-fc 静态

C:\Users\Administrator>
    
```

4. 主机 A、B、C、D、E、F 停止捕获数据，并立即在命令行下运行“arp-a”命令察看 ARP 高速缓存。

●ARP 高速缓存表由哪几项组成？

回答：三项。Internet 地址，物理地址和状态。

●结合协议分析器上采集到的 ARP 报文和 ARP 高速缓存表中新增加的条目，简述 ARP 协议的报文交互过程以及 ARP 高速缓存表的更新过程。

回答：工作过程如下：

1.ARP 模块接收到来自上层协议（IP）的数据报后，提取其目的 IP 地址。

2.检查目的 IP 地址在 ARP 高速缓存中是否有对应的项目。

3.若有，则使用该对应项的物理地址（MAC）作为数据包的目的 MAC 地址。

4.若没有，则广播发送 ARP 请求报文来获取目的 IP 地址所对应的物理地址，等待接收数据报。若没有收到 ARP 应答，则停止发送数据报；若收到 ARP 应答，使用应答中的物理地址（MAC）作为数据报的目的 MAC 地址，并将目的 IP 地址和应答中物理地址结合到一起，添加到 ARP 高速缓存。

### 练习 2 编辑并发送 ARP 报文（同一子网）

本练习将主机 A、B、C、D、E、F 作为一组进行实验。

1. 在主机 E 上启动协议编辑器，并编辑一个 ARP 请求报文。其中：

MAC 层：

目的 MAC 地址：设置为 FFFFFFFF-FFFFFF

源 MAC 地址：设置为主机 E 的 MAC 地址

协议类型或数据长度：0806

ARP 层：

发送端硬件地址：设置为主机 E 的 MAC 地址

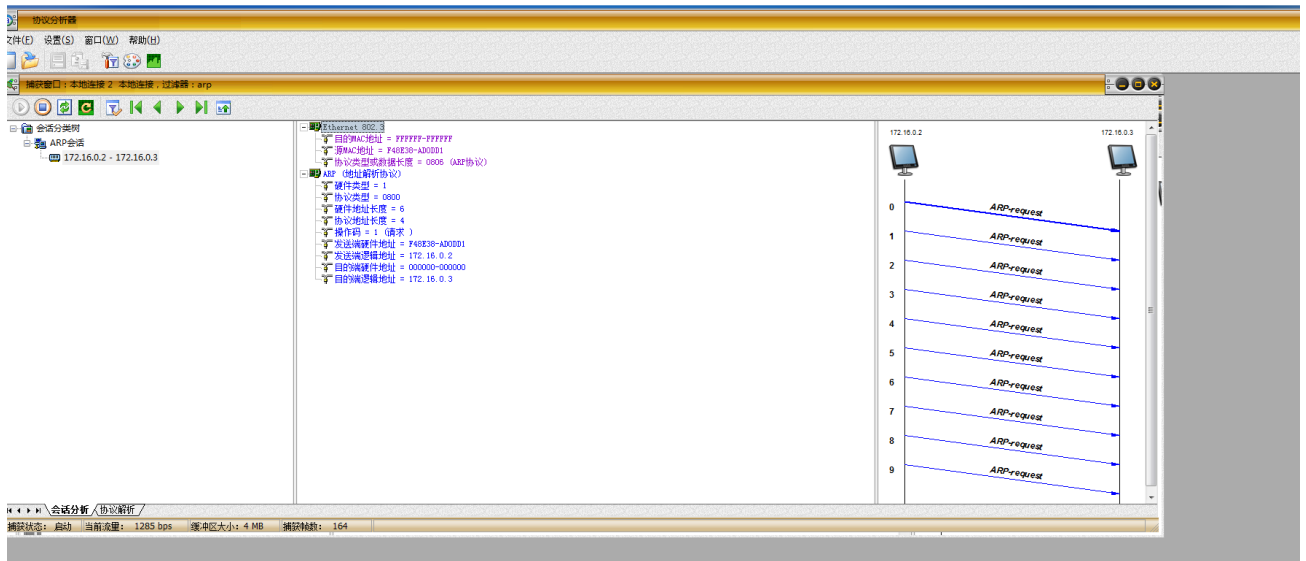
发送端逻辑地址：设置为主机 E 的 IP 地址（172.16.0.2）

目的端硬件地址：设置为 000000-000000

目的端逻辑地址：设置为主机 F 的 IP 地址（172.16.0.3）

2. 主机 A、B、C、D、F 启动协议分析器，打开捕获窗口进行数据捕获并设置过滤条件（提取 ARP 协议）。

3. 主机 B、E、F 在命令行下运行“arp -d”命令，清空 ARP 高速缓存。主机 E 发送已编辑好的 ARP 报文。



## 第 2 次实验 实验名称 地址解析协议 ARP

The image shows a Wireshark capture of ARP requests. The packet list pane shows 21 packets, all of type ARP (地址解析协议) with operation code 1 (Request). The packet details pane for packet 1 shows the following fields:

- 目的MAC地址 = FFFFFFFF
- 源MAC地址 = F4E38-AD0D1
- 协议类型或数据长度 = 0806 (ARP协议)
- ARP (地址解析协议)
  - 操作码 = 1
  - 协议类型 = 0800
  - 硬件地址长度 = 6
  - 协议地址长度 = 4
  - 操作码 = 1 (请求)
  - 发送端硬件地址 = F4E38-AD0D1
  - 发送端逻辑地址 = 172.16.0.2
  - 目的端硬件地址 = 000000-000000
  - 目的端逻辑地址 = 172.16.0.3

The packet bytes pane shows the following hex data:

```

00000000: FF FF FF FF FF FF F4 E3 8A D0 D1 08 06 00 01  08 06 00 01
00000010: 08 06 04 00 01 F4 E3 8A D0 D1 AC 10 03 00 00  00 00 00 00
00000020: 00 00 00 00 00 00 AC 10 03 00 00 00 00 00 00  00 00 00 00
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00  00 00 00 00
    
```

B 报文

The image contains two network diagrams illustrating the ARP process between two hosts, 0 (172.16.0.2) and 1 (172.16.0.3).

The top diagram shows the initial state where host 0 sends an ARP request to host 1. The packet details pane shows:

- 目的MAC地址 = FFFFFFFF
- 源MAC地址 = F4E38-AD0D1
- 协议类型或数据长度 = 0806 (ARP协议)
- ARP (地址解析协议)
  - 操作码 = 1
  - 协议类型 = 0800
  - 硬件地址长度 = 6
  - 协议地址长度 = 4
  - 操作码 = 1 (请求)
  - 发送端硬件地址 = F4E38-AD0D1
  - 发送端逻辑地址 = 172.16.0.2
  - 目的端硬件地址 = 000000-000000
  - 目的端逻辑地址 = 172.16.0.3

The bottom diagram shows host 1 responding to host 0 with an ARP response. The packet details pane shows:

- 目的MAC地址 = F4E38-AD0F87
- 源MAC地址 = F4E38-AD0D1
- 协议类型或数据长度 = 0806 (ARP协议)
- ARP (地址解析协议)
  - 操作码 = 2 (应答)
  - 协议类型 = 0800
  - 硬件地址长度 = 6
  - 协议地址长度 = 4
  - 操作码 = 2 (应答)
  - 发送端硬件地址 = F4E38-AD0F87
  - 发送端逻辑地址 = 172.16.0.3
  - 目的端硬件地址 = F4E38-AD0D1
  - 目的端逻辑地址 = 172.16.0.2

E 报文

The image contains two network diagrams showing the ARP process with IP addresses.

The top diagram shows host 0 (172.16.0.2) sending an ARP request to host 1 (172.16.0.3). The packet details pane shows:

- 目的MAC地址 = FFFFFFFF
- 源MAC地址 = F4E38-AD0D1
- 协议类型或数据长度 = 0806 (ARP协议)
- ARP (地址解析协议)
  - 操作码 = 1
  - 协议类型 = 0800
  - 硬件地址长度 = 6
  - 协议地址长度 = 4
  - 操作码 = 1 (请求)
  - 发送端硬件地址 = F4E38-AD0D1
  - 发送端逻辑地址 = 172.16.0.2
  - 目的端硬件地址 = 000000-000000
  - 目的端逻辑地址 = 172.16.0.3

The bottom diagram shows host 1 (172.16.0.3) responding to host 0 (172.16.0.2) with an ARP response. The packet details pane shows:

- 目的MAC地址 = F4E38-AD0D1
- 源MAC地址 = F4E38-AD0F87
- 协议类型或数据长度 = 0806 (ARP协议)
- ARP (地址解析协议)
  - 操作码 = 2 (应答)
  - 协议类型 = 0800
  - 硬件地址长度 = 6
  - 协议地址长度 = 4
  - 操作码 = 2 (应答)
  - 发送端硬件地址 = F4E38-AD0F87
  - 发送端逻辑地址 = 172.16.0.3
  - 目的端硬件地址 = F4E38-AD0D1
  - 目的端逻辑地址 = 172.16.0.2

F 报文

## 练习 3 跨路由地址解析（不同子网）


本练习将主机 A、B、C、D、E、F 作为一组进行实验。

1. 主机 B 在命令行方式下输入 `staticroute_config` 命令，开启静态路由服务。

```
C:\Users\Administrator>staticroute_config
[SC] ChangeServiceConfig 成功

SERVICE_NAME: remoteaccess
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE                : 2   START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE       : 0    (0x0)
        SERVICE_EXIT_CODE    : 0    (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x7d0
        PID                  : 352
        FLAGS                 :
```

2. 主机 A、B、C、D、E、F 在命令行下运行 “`arp-d`” 命令，清空 ARP 高速缓存。
3. 主机 A、B、C、D、E、F 重新启动协议分析器，打开捕获窗口进行数据捕获并设置过滤条件（提取 ARP、ICMP）。
4. 主机 A ping 主机 E（172.16.0.2）。



```
C:\Windows\system32\cmd.exe
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间<以毫秒为单位>:
    最短 = 1ms, 最长 = 2ms, 平均 = 1ms

C:\Users\Administrator>arp -a

接口: 172.16.1.2 --- 0xb
    Internet 地址          物理地址          类型
    172.16.1.4             f4-8e-38-ad-0d-c4 动态

C:\Users\Administrator>arp -d

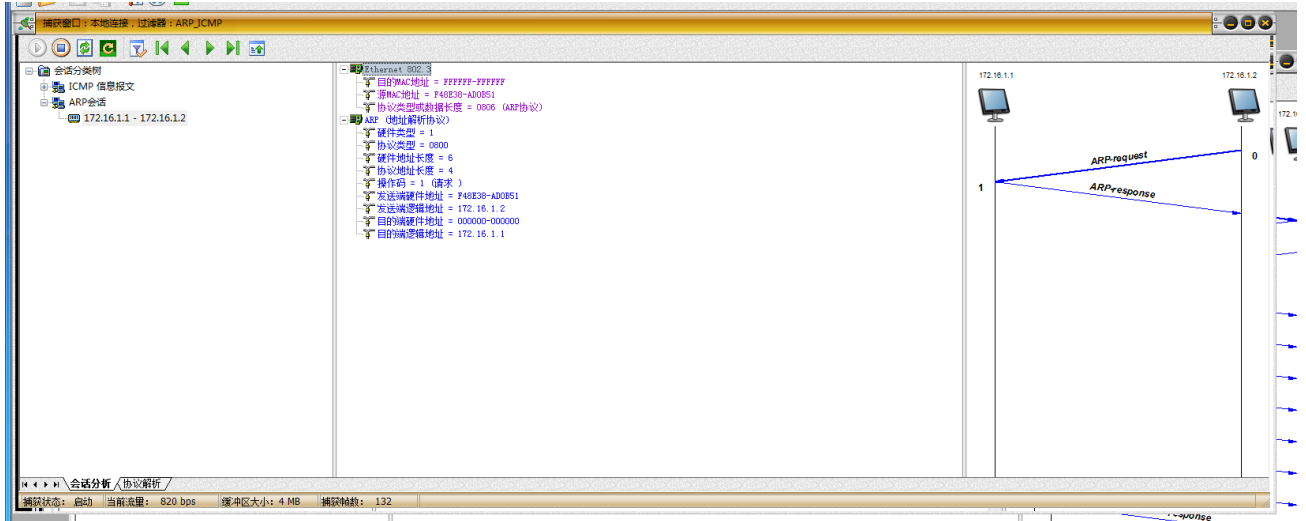
C:\Users\Administrator>ping 172.16.0.2

正在 Ping 172.16.0.2 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

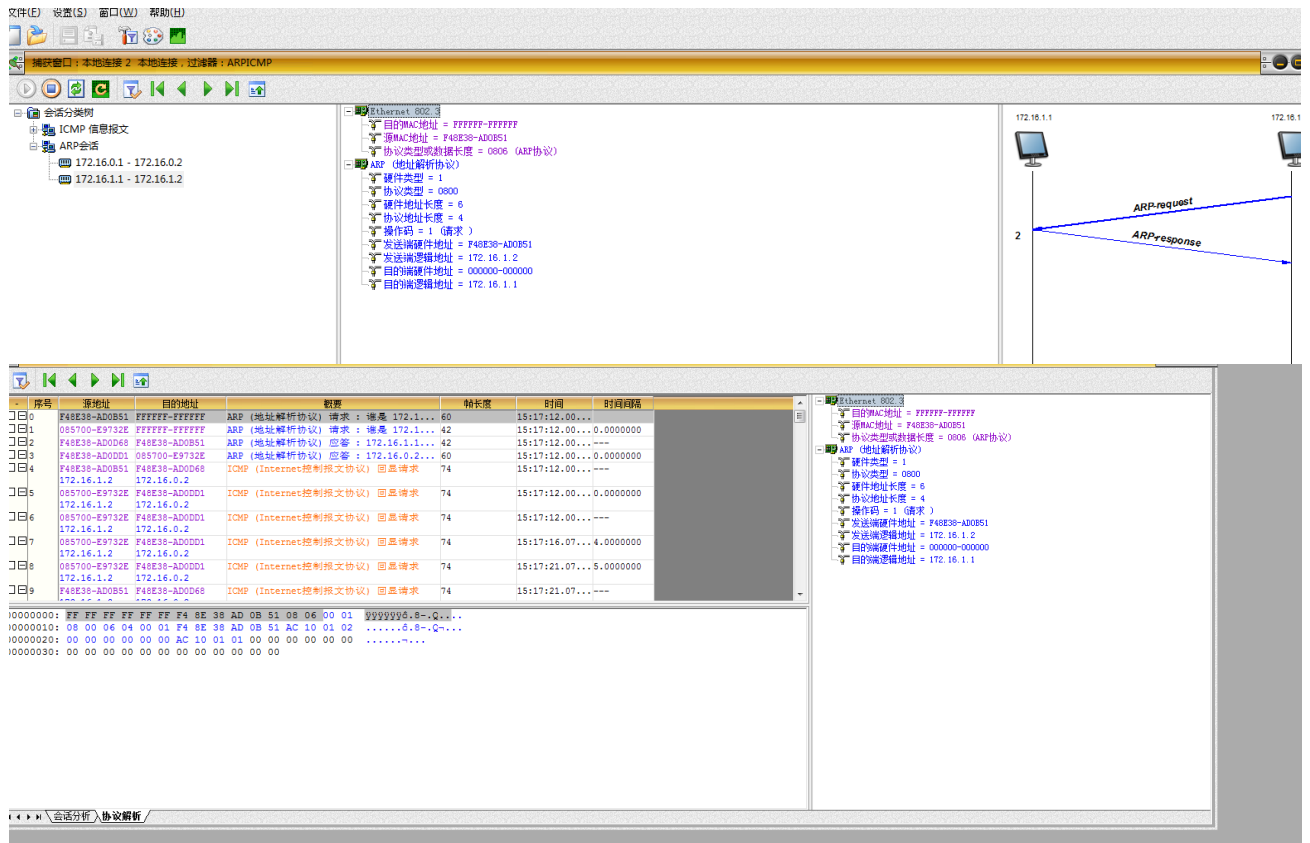
172.16.0.2 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\Administrator>
```

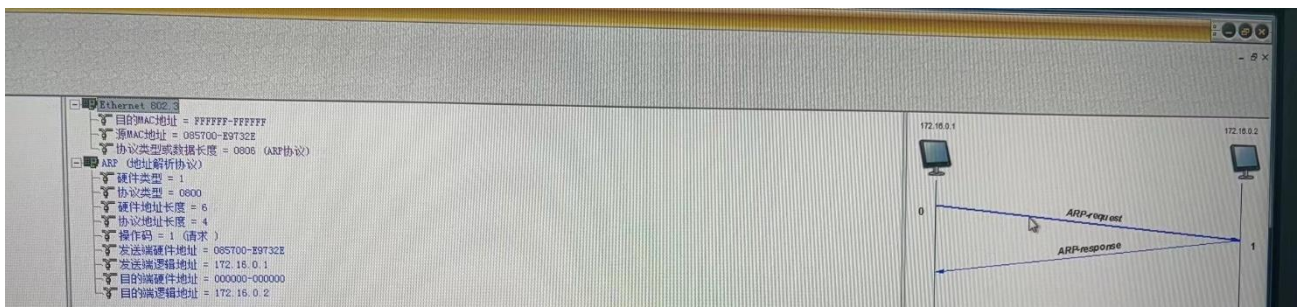
5. 主机 A、B、C、D、E、F 停止数据捕获，察看协议分析器中采集到的 ARP 报文，并回答以下问题：



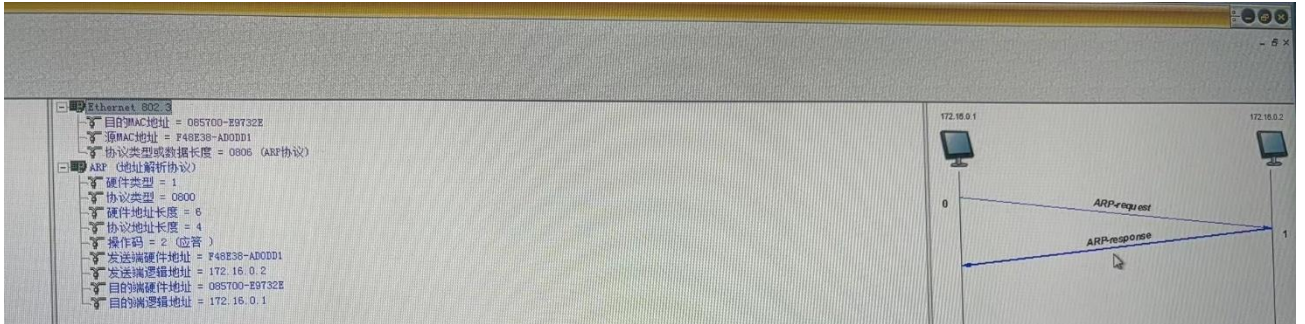
A 报文



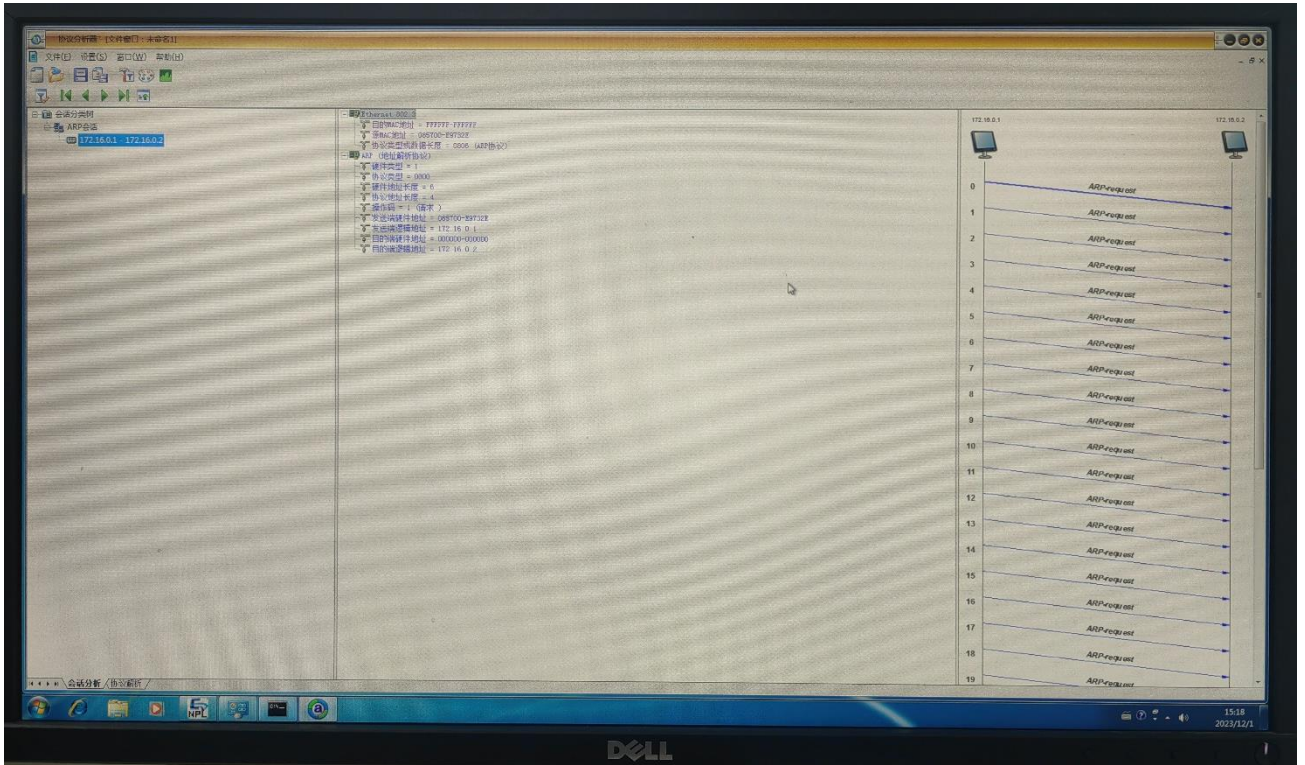
B 报文







E 报文



F 报文

●单一 ARP 请求报文是否能够跨越子网进行地址解析？为什么？

回答：不能。因为 ARP 请求是广播包，只能在单个子网内广播，而无法跨子网。换句话说，即使发送者知道目标 IP 地址，但是它无法通过 ARP 来获取目标 MAC 地址，因为 ARP 请求无法跨子网广播。

在这种情况下，通常需要使用其他技术如代理 ARP 或者 NAT（网络地址转换）等来帮助实现跨子网的地址解析。

●ARP 地址解析在跨越子网的通信中所起到的作用？

回答：由于 ARP 报文的存活空间只限在子网内，因此 ARP 本身无法跨越不同网段。在跨越子网的通信中，ARP 地址解析的作用主要是解析网关的 MAC 地址。当数据要发往外部网络时，通常是首先使用 ARP 请求网关路由器的 MAC 地址，之后将数据发往网关路由器，由网关路由器进行转发。

6.主机 B 在命令行方式下输入 recover\_config 命令，停止静态路由服务。

## 五、 实验结果总结

思考问题:

1.练习 2: 哪些主机收到了 ARP 请求包, 哪个主机给出了 ARP 响应包?

回答: B 和 F 收到了 ARP 请求包, F 给出了 ARP 响应包。

2. 练习 2: 主机 A、C、D 是否收到 ARP 请求包, 为什么?

回答: 主机 A、C、D 没有收到 ARP 请求包。因为 ARP 请求包是通过广播实现的, 并不能跨越子网, 主机 A、C、D 和主机 E、F 不在同一个子网中。

3. 练习 3: 哪些主机收到了 ARP 请求包, 哪台主机给出了 ARP 响应包?

回答: B、C、D、F、E 收到请求包, E 给出了响应包。

4.比较 ARP 协议在同网段内解析和跨网段的解析有何异同点?

回答:

相同点: 无论是同网段内还是跨网段, ARP 的作用都是一样的, 都是用来广播发送 ARP 请求报文, 尝试解析 IP 地址到 MAC 地址的映射关系。

不同点: 在同网段内, ARP 请求能够被同网内的所有主机接收到。当一台主机发送 ARP 请求后, 只有符合 ARP 请求中 IP 地址字段条件的主机才会进行应答, 将自身的 MAC 地址发送给发送 ARP 请求的主机。

跨网段解析: 当 ARP 请求是跨网段时, 无法直接将 ARP 请求发送到目标网络。这时需要借助路由器, 将 ARP 请求发送到目标网络。

5.ARP 数据包的长度是固定的吗? 试加以解释。

回答: 不是固定的。协议地址长度、发送端物理地址、接收端物理地址是可变长度的。

6.试解释为什么 ARP 高速缓存每存入一个项目就要设置 10-20 分钟的超时计时器。这个时间设置得太大或太小会出现什么问题?

回答: ARP 高速缓存每存入一个项目就设置 10-20 分钟的超时计时器的原因主要是为了维护 ARP 缓存的一致性和有效性。如果超时时间设置得太大, 那么 ARP 缓存中可能会保留很多过时或无效的条目。这会导致 ARP 缓存变得非常大, 消耗大量的内存资源。此外, 如果缓存中存在过时条目, 还可能影响网络的性能, 因为发送主机可能会尝试发送数据到已经不存在或已经移动了的主机。如果超时时间设置得太短, 那么 ARP 缓存中可能会频繁地清除条目, 导致网络性能下降。此外, 如果条目被频繁地清除和重建, 还可能会导致网络的不稳定。

7.至少举出两种不需要发送 ARP 请求数据包的情况。

回答: 当 ARP 缓存表中已经含有要解析的条目, 或者目标主机的 IP 地址和本机在同一网段内。